CONTINUOUS AUTHENTICATION USING

BEHAVIOURAL BIOMETRICS

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

SHAILESH MISHRA (161458)

UNDER THE SUPERVISION OF

RIZWAN UR REHMAN

TO

Department of Computer Science & Engineering and Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT, SOLAN, HIMACHAL PRADESH,173234

ACKNOWLEDGEMENT

Any serious and lasting achievement cannot be achieved without the help, guidance and co-operation of numerous people involved in the work.

First and foremost, I would like to express my gratefulness to Prof. Dr. Samir Dev Gupta Professor and Head of Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology for providing us the opportunity to carry out this project as our final year project. It gives us immense pleasure to express my gratitude and thanks to Rizwan Ur Rehman, Assistant Professor, Department of Computer Science & Engineering and Information Technology, for not only imparting his knowledge but also his constant supervision, advice and guidance throughout the project, without which this project wouldn’t have been possible. I would also like to thank all other department faculty at Jaypee University of Information Technology. Not only did they taught us and made us capable enough to undertake this project but were always there at the need of the hour and provided with all the help, facilities and co-operation, which was required towards the completion of the project. A special mention to Ravi Raina Sir and Sanjeev Kumar Sir who assisted our project lab and guided us towards all the minor issues. Last but not the least, I would like to express our thanks to our parents and family members for their support at every step of my life.

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled **"CONTINUOUS AUTHENTICATION USING BEHAVIOURAL BIOMETRICS"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2019 to May 2020 under the supervision of **Rizwan Ur Rehman**(Assistant Professor, Computer Science& Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Shailesh Mishra (161458)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Rizwan Ur Rehman

Assistant Professor

Department of Computer Science & Engineering and Information Technology,

Jaypee University of Information Technology

Dated:

TABLE CONTENT

LIST OF FIGURES

LIST OF ACRONYMS & ABBREVIATIONS

KD-Keystrokes Dynamics

SVM-Support Vector Machine

SA-Static Authentication

CA-Continuous Authentication

CBAS-Continuous Biometric Authentication Schemes

FAR-False Acceptance rate

FRR-False Rejection rate

EER-Equal error rate

UDKL-Up Down Key Latency

DUKL-Down Up Key Latency

UUKL-Up Up Key Latency

DDKL-Down Down Key Latency

DT-Dwell Time

FT-Flight Time

ANGA-Average Number of Genuine Actions

ANIA- Average Number of Impostor Actions

ABSTRACT

Behavioural biometrics is the field of study which relates to the measure of uniquely identifying and measuring the patterns in human activities. Computer security plays a vital role as most of the sensitive data is stored on computers. Keystrokes Dynamics is a technique based on human behavior for typing anything on keyboard. Whenever any user logs into the system, username and password combinations are used for authenticating the users. The username is sometimes not secret, or predictable and the imposter acts as the official user to guess the password also because of simplicity of password(sometime), the systems are prone to more attacks. We also demonstrate a new way to perform continuous authentication using Mouse Dynamics as the behavioral biometric modality. In the proposed scheme, the user will be authenticated per mouse event performed on his/her system. In this case biometrics provide secure and convenient authentication. Our system uses a Support Vector Machine i.e. SVM, which is one of the best known classifications and regression algorithm for this purpose. Support Vectors i.e. SV that fall under different regions is separated using hyper planes, linear as well as non-linear. Researchers have proved that SVM is convenient for usage as it will converge to the best possible solution in very less time. The False acceptance Rate and False Rejection Rate is calculated at the final result.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Fraud and Impersonation are two main causes that pose threat to data, digital network and computer system security. Many web-based authentication systems have been proposed to safeguard commercial transactions and to secure the data. Ideas such as account username and password, IP address filtering, message digest authentication, etc. are the popular ones. It can be assumed that these systems will never be all perfect, one can only make them better with more and more security. For example, if a user goes with a weak password then it can easily be cracked. In the hope of improving on this, many research studies have been done to process user input data in such a way that it can be used as a form of authentication. One out of these promising approaches has been Keystroke biometrics which refers to the habitual patterns or rhythms a personal exhibit whereas writing on a keyboard device.

Compared to alternative biometric schemas, keystroke has the primary advantages that:

1. No external hardware like scanner or detector is needed. All that is wanted is a keyboard.

2. **The “rhythm” or the “pattern” of the users is a very reliable statistic.**

3. It can easily be deployed in addition with existing authentication systems.

The keystroke authentication approach has been divided into two most common approaches. First approach concentrates on the “static verification” where the user is provided with a default set string which gets displayed and the user has to type that same string in the space provided so that the user’s typing pattern can be calculated from the input. Second approach is called “free-text” dynamics in this approach the user who is assumed remember the username and password and hence the input is taken as the typing pattern out of the username and password. The safer way is the second one because it is an additional layer that only the true user will remember the typing pattern for specific username and password unlike the static on which has a constant string, Hence, it is known as Continuous Authentication.

Protecting the device form any unauthorized access is very important, we use username and password-based authentication, also a two-factor authentication where one can ask for an OTP but that is not a continuous authentication. This is also a proof of identity throughout the login process but it is not a continuous process after login wherever user uses a keystroke, known as static Authentication and the user is supposed to be a legitimate user throughout even after the login process is over. But Continuous Authentication (CA) is where the legitimacy of a user is constantly monitored based on the biometric signature left which is the typing patterns of the user on a particular device. If a doubt comes about the legitimacy of any user then one can revert back to static authentication, the system will get locked, and the user has to revert to SA. Continuous authentication is not an alternate security but it is an additional layer of security. Hence if the CA based system detects any user as an imposter then the system should get locked up to avoid any unauthorized access of the system or any important files. The basic need or the foremost need for having a secure system is preventing unauthorized access to the system or any files and if an access granted somehow to any imposter then the system should be able to take the necessary action and should be quick in that, here the action is to logout the imposter as quick as possible. A question arises about the token of the access that should or should not be provided to the user of the access of the system. As in SA the token is provided. Knowledge based systems are the one that will tokenize the user about the attempt will only disturb the user once having to type the password, whereas possession-based systems aren't effective for users that don't take away their token once deed the system unattended. Besides, a token will provide the false user a knowledge of the false attempt and will only weaken the system as the imposter would know that false attempt. Hence continuous authentication can easily be done using the biometric system using keystroke dynamics.

In the prototype model the user typing pattern would be compared with the typing pattern of the actual user and if it matches, then user can log in otherwise the current user will be logged out. The difference between the normal login system and the biometric system is that the biometric system will mostly be dealing with the uniqueness of the user’s biological features here the typing pattern or the user interaction with the keyboard and these provide some statistical features that are unique for a user. Once the user logs into the system then the user has complete access to the system files and applications and sometimes other users can also login after the admin access is provided and this continues until the user logs out. This can be not of a big threat when there are not many important data in the system but sometimes this can lead to session hijacking where session logins are required. But in CA based system the user will be checked continuously when the system is kept idle or some ambiguous typing pattern is detected. Continuous Biometric Authentication Schemes i.e. CBAS is built around the biometrics features supplied by the user behavioral characteristics and continuously checks the identity of the user throughout a session.

We found that the current research on continuous authentication reports the results in terms of EER or FAR and FRR over either the whole test set or over chunks of a large, fixed number of events. This is then in fact no longer continuous authentication, but at best periodic authentication. With the ever-expanding interest for progressively secure access control in a large number of the present security applications, customary techniques, for example, PINs, tokens, or passwords neglect to stay aware of the difficulties introduced in light of the fact that they can be lost or taken. On the contrary hand, biometrics dependent on "who" the individual is or "how" the individual acts present critical security progression to address these new difficulties. Among them, keystroke elements give a characteristic decision to make sure about "secret key free" PC get to.

User’s typing rhythm can be seen as unique as a user’s signature. These rhythms are dependent on the mental conditions but if made habitual and necessary as in keystroke dynamics, then the user will have to remember or become habitual with a pattern. Truth be told, as right on time as the 19th century, transmit administrators like telegraph operators could perceive each other through their particular tapping styles. This recommends keystroke elements contain adequate data to fill in as a biometric identifier. Our framework for keystroke elements has various stages for client verification.

1.User authentication details
2.Training the system
3.Calssification using an algorithm
4.FAR and FRR calculation

1.1.1 Keystroke Dynamics

The starting point of keystroke biometrics can be accepted to in 1880 when individuals had the option to distinguish their associates with their particular composing sounds. Customary verification frameworks depend on the utilization of a mutual mystery, a secret key or pin-number that is just known to the framework and the client. The framework confirms the character of the client by mentioning the mutual mystery and contrasting the client reaction with the normal reaction, tolerating the client personality upon a match. In such a framework, the security is subject to the common mystery staying covered up, as anybody realizing the mutual mystery could effectively imitate that client.

There are various strategies to improve the security of these frameworks and most include the utilization of some type of particular equipment, for example, a unique mark sensor, two-factor key generator, advanced marks dependent on awry cryptography with the private key put away on a brilliant card and requiring a PIN code, for example, the BankID framework utilized by the significant banks in Sweden, iris scanner and camera for face acknowledgment. Keystroke biometrics can likewise be sent in a current domain without the requirement for extra equipment.

The objective of keystroke biometrics is to distinguish or confirm clients dependent on the individual composing attributes, for instance the planning of every keystroke, the weight applied when composing and on account of cell phones the direction of the gadget, accelerometer information, size of touch and area of touch. Timing information for keystroke biometrics are made by recording when each key is squeezed and when it is discharged. From the planning information, one can remove various highlights, for example, latencies. The highlights that are normally utilized are abide time, Flight time and digraph, which are in this manner alluded to as the basic highlights. Abide time is characterized as the time between the key is squeezed and its discharge. Flight time is characterized as the time between the key is discharged and the ensuing key is squeezed. The digraph is characterized as the time between key presses of two ensuing keys.

- TD and KD is the point by point timing data that depicts precisely when each key was squeezed and when it was discharged as an individual is composing at a PC keyboard.

- The recorded keystroke timing information is then prepared through an extraordinary neural calculation, which decides an essential example for future examination.

- Static verification
- Only typing rhythm from login time is taken.
- Authentication just at login time and not after for example not consistent.
- Dynamic verification
- pattern paying little mind to the composed content of username and secret phrase.
- A consistent and time-to-time check for nonstop confirmation

<u>Biometric System</u>

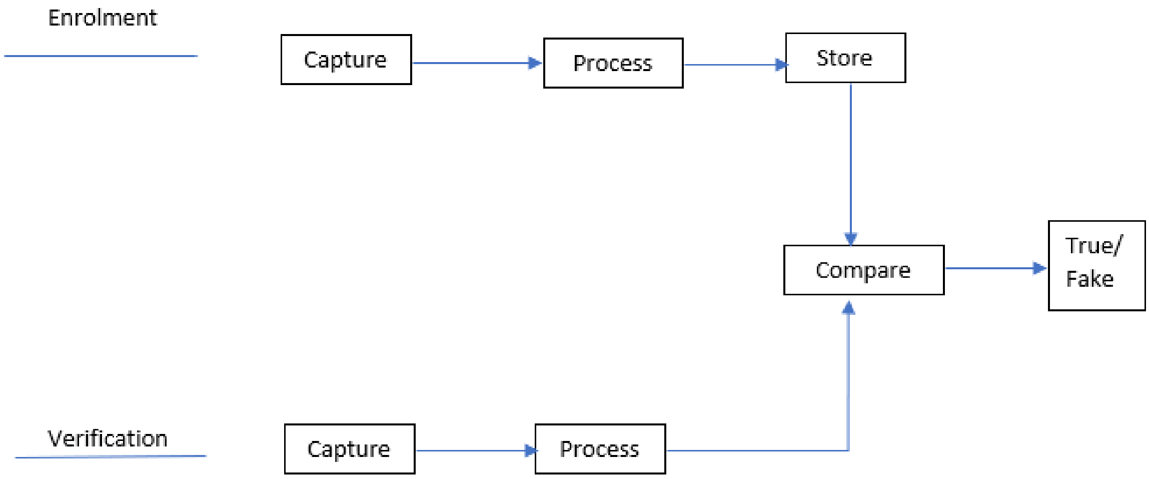


Figure 1: The different biometric System stages and its architecture.

There are many features of keystroke dynamics such as:

- Typing Speed-No. of words typed in a minute
- Flight time-Time elapsed between keystrokes
- Dwell time-Time elapsed in during key-press
- Method for error correction-Frequent backspace or select-delete
- Characteristic error frequency-Typing error rate.

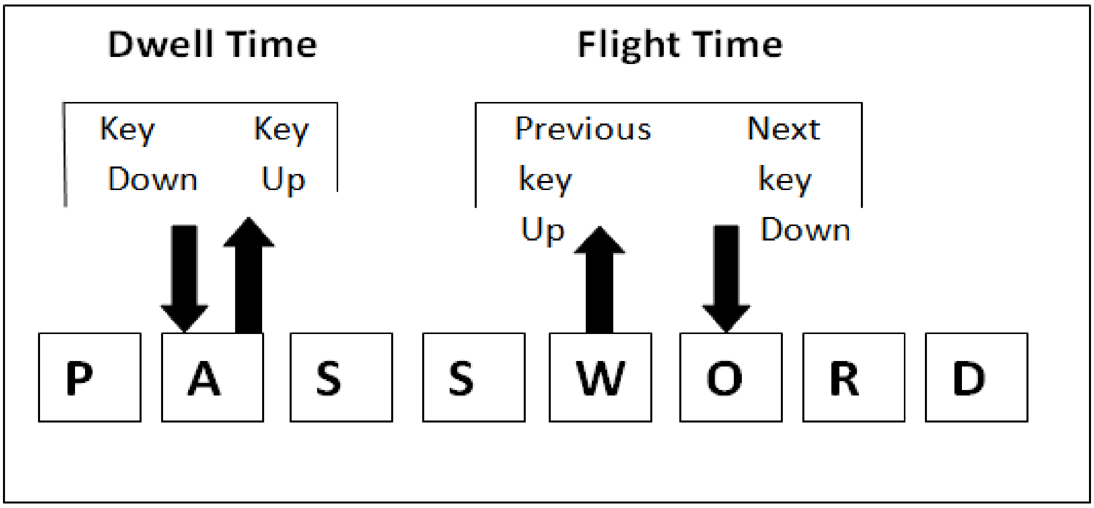


Fig 2: Dwell time, Flight time

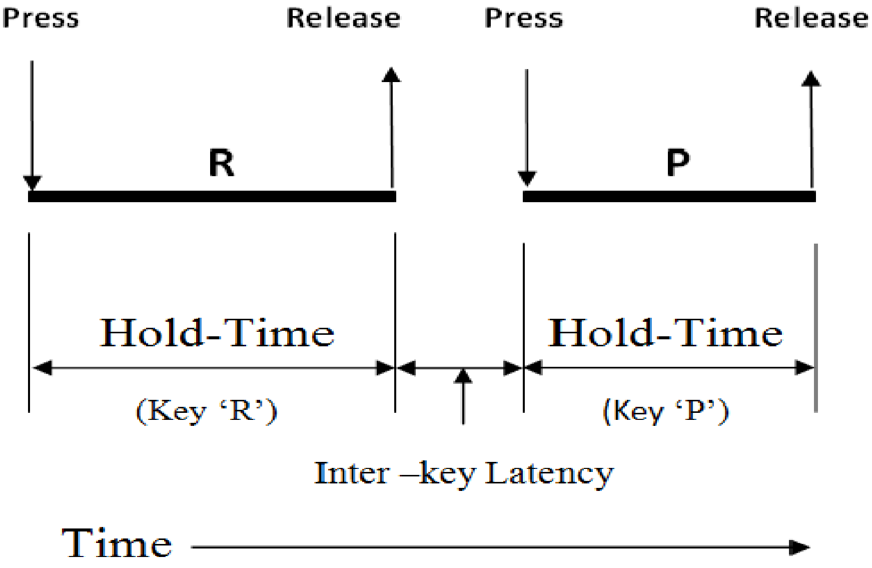


Fig 3: Inter key latency

<u>Matrices for Keystroke Dynamics</u>

Many classifiers unit of measurement procurable for Keystroke dynamics until date, thus these models are validated based on security metrics like False Acceptance Ratio (FAR), False Rejection Ratio (FRR).

1. False Rejection Ratio (FRR): The fraction or percentage of times a true user of the system is logged out and is recognized as an imposter. This metric should be low for a successful model.
2. False Acceptance Ratio (FAR): The fraction or percentage of times a false user or an imposter is allowed to log in. This metric should also be very low for a successful model.
3. FRR is the quantitative relation of number of false rejections divided by total number of genuine match attempts. Thus, FRR provides the quantity of real users.

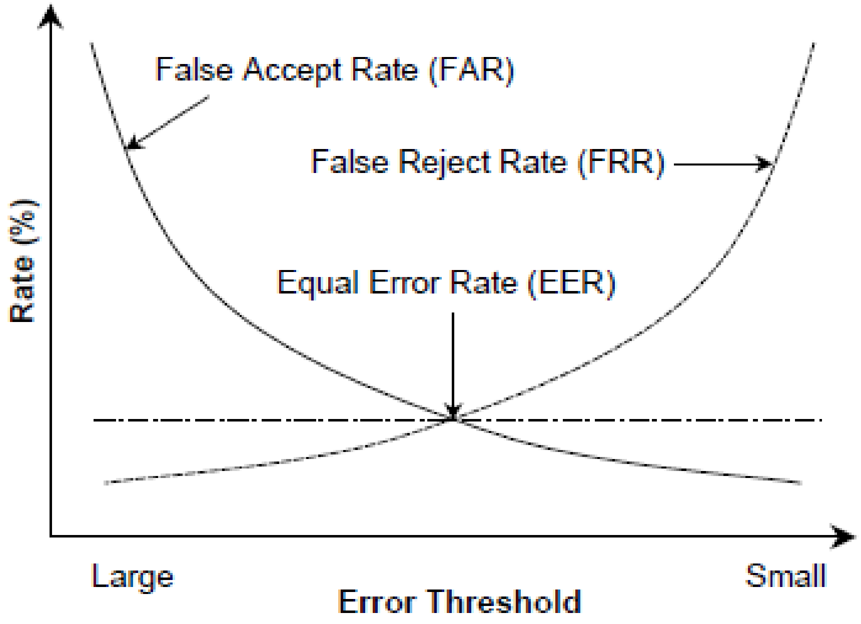


Fig 4: The graph showing error threshold

Features/Metrics as follows:

1.1.1.1 Typing Speed-In Typing Speed we calculate the number of words press by the user per minute.

- Finger placement i.e. the place where the finger is placed on the key or even the angle of the fingers when pressing the key This feature is not necessary but can only be assumed.
- Pressing time i.e. the timestamp recorded when the key is pressed.
- Releasing time i.e. the timestamp recorded when the key is released.

1.1.1.2 Flight Time-This is the time elapsed between the previous key-up and current key-down and farther the keys more is the time and is to be unique for a user.

1.1.1.3 Dwell Time- This is the time elapsed on the same key i.e. the time between a key-down and key-up on the same key. This is also unique for a user.

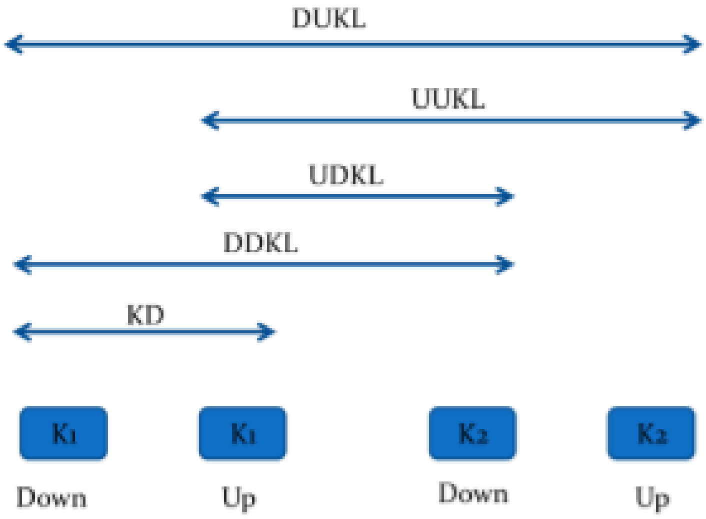


Fig 5: Flight time and Dwell time calculation.

The different features are defined as the building block for the behavioral biometric using keystroke dynamics. Here flight time is defined as the time elapsed between the previous key-down and the next key-up and if the keys are farther than this might take a longer time for a user who is not habitual with typing the current user’s password in the same way and speed. Similarly, the other feature Dwell Time is the time elapsed between the current key-up and key-down and the last feature is the typing speed of the user. There can be additional features to increase the system efficiency.

There exists JavaScript function for calculation of the various features using time function.

SUITABILITY OF KEYSTROKE DYNAMICS

Keystroke dynamics are not that reliable as many other biometric systems that uses fingerprint or face detection but it is very useful, mainly we classify them as following seven criteria:

Universality-Keystroke dynamics just requires a keyboard and hence is universal and can be used by people who want to secure their system using behavioral biometrics.

Uniqueness-Again, behavioral biometrics is not that strong and we cannot say that the user will have a unique pattern all the times unlike the other physiological biometric factors. Physiological biometrics are very unique and hence will have a very low False rejection ratio but in keystroke dynamics one can try to decrease it by increasing the no. of extracted features. Therefore, we cannot make keystroke dynamics authentication as the primary securing layer but it definitely can be secondary or tertiary.

Permanence-User’s typing Pattern will vary as time grows usually user’s typing speed increases and the multiple attempts for password change will update the typing rhythm and hence, we cannot say the that the users typing rhythm can be kept permanent but instead it has to be updated as time passes.

Collectability-Keystroke dynamics is universal and can be used anywhere where there is a work involving keyboard and security. There is no requirement of any more hardware as is many other biometric systems like sometimes camera is required in surveillance and fingerprint scanner is required but in keystroke dynamics only keyboard is required which is handy with a computer and data collection and feature extraction is easy.

Acceptability-It maybe against the law, taking users input data or recording without permission of the authority here the user. In some countries the consent is a must while recording data in to the system otherwise it is a privacy breach. Given that the imposter is not logged in and the data will not be retained but still it will be the case of privacy breach depending on the country. Hence, it requires a legal advice from the owner before implementing or even experimenting keystroke dynamics or like it which records data in the system memory.

Circumvention-It is almost impossible to generate this data automatically or match the data automatically but it is not impossible. Like it can be achieved using certain keyloggers hidden in the system hence, the software must be installed keeping these things in mind.

Performance-Sometimes a true user might be not in mood or out of fatigue can not figure out the pattern and might differ from actual pattern hence the actual user would not be able to login out of very genuine reasons hence, it is not that successful as other physiological biometrics and this have a larger False acceptance ratio compared to them.

1.1.1.4 **Method for error correction-**The Frequent backspace or select-delete means to Correct the error what user mostly press the key backspace or select/delete.

1.1.1.5 **Characteristic error frequency-**The Typing error rate at which the user meets with an error.

1.2 Problem statement

The keystroke includes that are as of now utilized by social biometrics frameworks are abide time, key flight and digraph, which are extricated from the timestamps of the keystroke occasions. A restriction of the present techniques for highlight extraction is that they are straight blends of the timestamps doesn't rearrange possible non-direct connections between the keystrokes. Hence, any fundamental non-straight structure is left to the classifier to find. Highlight learning can be utilized to all the more precisely catch the basic structure of the information, which can improve order exactness or taking into consideration the utilization of progressively straightforward classifiers. The advantage of a more straightforward classifier is that it can work in an asset compelled condition, for example, a cell phone or in an internet browser.

The goal of this work is to explore a couple of various techniques for include extraction and test whether these strategies improve the grouping exactness or empower preparing with less examples, and furthermore to analyze the outcomes got how they stack facing.

There are many conventional features and methods. In particular the aims on the:

- It investigates feed-forward neural networks for use as classifier.

- It evaluates the classification accuracy while using pre-training and compare it to conventional training.
- It evaluates the two training methods for various width and depth settings of the neural network.
- It investigates auto encoders for use in feature extraction, exploiting the auto encoders' capacity to learn useful properties about the data.
- It can study any behavioral similarities between multiple users.
- It explores the structure in the information that is explicit for a solitary client.

1.3 Objective

The target of this venture is to build up a model for a safe web application which utilizes pre-put away information about clients composing examples and mouse use examples and use it for persistent verification. Conduct biometrics is the field of particularly recognizing and estimating the examples in human exercises and their cooperation with the framework. Keystroke and mouse elements is strategy in which we quantify and dissect one's composing designs on console and mouse separately. In this venture we will utilize a few highlights of keystroke elements to extricate the information of the client and will locate the validated client.

1.4 Methodology

The proposed prototype model is based on keystroke Authentication. It is a type of behavioral biometric authentication system in which one user's typing rhythms are compared with the others. There are different parameters for measuring keystrokes of desktop users which can be categories in two main parts as attributes measured and performance metrics.

There are parameters for measuring mouse metrics that are mentioned below.

After measuring these metrics from the users from the login pages, the different metrics are stored as csv file in the memory. Then we apply SVM for calculation of threshold of all the metrics, which is referred at the time of verification and Authentication of users by measuring FAR and FRR.

Attributes Measured for keystrokes Dynamics are:

i. Latencies between successive keystrokes

ii. Keys hold durations

iii. Method of error correction

iv. Typing speed

v. Characteristic error frequency

Performance Metrics are two, that are described as:

i. False Rejection Ratio (FRR): The fraction or percentage of times a true user of the system is logged out and is recognized as an imposter. This metric should be low for a successful model.

ii. False Acceptance Ratio (FAR): The fraction or percentage of times a false user or an imposter is allowed to log in. This metric should also be very low for a successful model.

1.5 Organization

Chapter 1: Introduction about the project and mention of what the project does and what I am trying to accomplish with this project and how will it help a user.

Chapter 2: Literature survey for the project. This includes the various project reports on the previously made projects on biometric based authentication and machine learning and its applications.

Chapter 3: System Development, here I have mentioned the main architecture of the project and how the things are linked to each other and what platforms and overview of algorithms I have used in building this project.

Chapter 4: Performance Analysis of the project. Here I have mentioned the results and how the system is performing and what is the success and failure rate.

Chapter 5: Conclusion. Here I have mentioned the outcomes and the future scopes of the project and what else can be done and what are the applications.

CHAPTER - 2

LITERATURE SURVEY

Summary of papers

Title **Keystroke Dynamics in a general setting**

Authors Terence Sim and Rajkumar Janakiraman, School of Computing, National University of Singapore, Singapore 117543

{rajkumar,tsim}@comp.nus.edu.sg

Springer-Verlag Berlin Heidelberg 2007

Year Of Publication **2007**

Publishing House S.-W. Lee and S.Z. Li : ICB 2007, LNCS 4642, 2007. Springer-Verlag Berlin Heidelberg 2007

Summary Behavioral biometrics using Keystroke Dynamics can be treated as similar to the user's signature. Using username and password is very common, some software or application also use OTP for security. This project talks about the usage of keystroke dynamics metrics and features for authentication during internet browsing, email or message writing etc. They differentiate using two classification of processes as continuous authentication and one-time authentication. To use keystroke dynamics, they introduced a term called Goodness measure that computed the word quality. Using special characters and numbers in password also became a special feature for storing metrics and analyzing patterns more appropriately. Some common list of strings was used for training the model and eventually a specific list of string for specific individual were given for analyzing their keystroke patterns. During verification stage the same list of strings were used or specific users.

Summary of papers

Title **Authentication Method through Keystrokes Measurement of Mobile users in Cloud Environment**

Authors Mahnoush Babaeizadeh, Majid Bakhtiari, and Mohd Aizaini Maarof

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia

e-mail: mahnoush.b@gmail.com, bakhtiari@utm.my, aizaini@utm.my

Year Of

Publication 2014

Publishing

Details International J. Advance Soft Compu. Appl, Vol. 6, No. 3, November 2014 ISSN 2074-8523

Summary When a user wants to access cloud services using internet then authentication can be done using Mobile Cloud Computing. Username and password verification are common on most of the platforms but, while using a device that can receive an OTP then mobile cloud computing uses OPT also. Keystroke Dynamics is a new type of authentication method that this project uses for Mobile Cloud Computing. Apart from Keystroke Dynamics, this project uses cryptography for analyzing patterns and cryptography is a new and efficient form of security that enables addition of security of the system. The project used simulation based on Junit packages and found their project to be very efficient and the imposter was not able to log in 97.33% of times. This was because keystroke patterns of different users will be different for the same set of passwords. This proved to be very efficient and it became difficult to copy typing patterns and hence, this method enhanced the security of the system for authentication in Mobile Cloud Computing.

<u>Summary of papers</u>

Title **Adaptive approaches for keystroke dynamics**

Authors Paulo Henrique Pisani, Ana Carolina Lorena, André C. P. L. F. de Carvalho

Universidade de São Paulo, Brazil

Year Of Publication 2015

Publishing Details 2015 International Joint Conference on Neural Networks (IJCNN) INSPEC Accession Number: 15503795

Summary In this study, the keystroke dynamics is used for recognizing patterns and then classifying these patterns using various classification algorithms hence, it can be said that this project is a one classification problem. For generating a model certain examples from a particular class were used after the classification is completed. The true user typing patterns were used initially for training the models and classification algorithms were used on these datasets. During the testing time the other dataset was used and that classified as the imposter dataset which proved to be useful. When generating an output for a particular true user the training data streams were kept apart and were not used at time of verification. So, this was just a simulation-based technique. Once classified there are no tokens or labels as a trusted or non-trusted dataset but simply a classified data was induced in the algorithm for analyzing the true and false user.

Summary of papers

Title **On Mouse Dynamics as a Behavioural Biometric for Authentication**

Authors Zach Jorgensen and Ting Yu

Department of Computer Science North Carolina State University Raleigh, NC 27695

{zjorgen,tyu}@ncsu.edu

Year Of

Publication 2011

Publishing

Details ASIACCS '11 Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ACM 978-1-4503-0564-8/11/03

Summary In this paper the author managed an experiment that was actually done on people who volunteered for the experiment and tried to resolve a query that the difference in the mouse usage patterns of the users were due to some environmental aspects or computational problems or users fatigue generation or some other aspects. The study found out that there are minute differences but the differences are noticeable and hence, can be used for training and verification. The dataset for mouse feature should not be small as there are not very features different from others. Hence, large dataset is required for verification and classification before reaching to a decision of true or false user. The project explained that the training and verification data for the same user should be collected from two or more different pointing devices and existing techniques were not able to classify easily. This finding suggests that mouse dynamics may not be a good choice for authentication in web-based applications or other remotely accessed resources.

Summary of papers

Title **User Authentication Through Mouse Dynamics**

Authors Chao Shen , Zhongmin Cai, Xiaohong Guan, Youtian Du, State Key Laboratory for Manufacturing Systems, Xi'an Jiaotong University, China

Roy A. Maxion, Dependable Systems Laboratory, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

Year Of Publication 2012

Publishing Details IEEE Transactions on Information Forensics and Security

Volume: 8 , Issue: 1 , Jan. 2013

Summary In this project report, we concentrated on those difficulties that looked by mouse-elements based client confirmation. At that point separation-based element development and parametric eigenspace change are applied to acquire the overwhelming element segments for proficiently speaking to the first mouse include space. toward the end, a one-class grouping method is utilized for playing out the client validation task. The dataset for mouse highlight ought not be little as there are not very highlights unique in relation to other people. Subsequently, enormous dataset is required for confirmation and grouping before coming to a choice of valid or bogus client. Enormous scope tests have shown the legitimacy of the proposed approach, with a bogus acknowledgment pace of 8.74%, a bogus dismissal pace of 7.69%, and a verification time of 11.8 seconds. These outcomes recommend that mouse elements can give a noteworthy improvement to conventional verification frameworks.

CHAPTER 3

SYSTEM DEVELOPMENT

The prototype model is developed by taking user's keystroke patterns as a dataset that can be user later for comparison and classification of the imposter from the actual user. It has already been discussed that typing pattern of a user might vary with the environment conditions and fatigue generation or other things but, it can also be seen as an additional pattern that has to be remembered by the user for logging in. At least they will have to remember that there is an additional layer for security. Hence, the genuine user keystroke pattern will differ from the imposter most of the times. The dataset is split in halves for the imposter training and also some training for the imposter is done using a different user than the true user.

3.1 Trust Model

Trust model is built by calculating the trust score for the user with the help of its parameters. There is a term defined that calculates the probability of the genuineness of a user and there is term called classifier score. The trust score depends on the probability of the occurrence of a particular event (here it is the occurrence of the pattern), if the pattern matches then the probability will be high and so will be the trust score. The parameter of the trust model changes as the classifier score of the genuine user and the imposter. But I have used the trust model in a different way. The plots are shown in the end using the data points of the genuine user and also the imposter. The genuine user plots almost lie on top of each other while that of an imposter differ from each other very much. This makes easy for the classification algorithm to note the difference in the patterns and classify it as an imposter or true user. In the end the False Rejection ratio and False acceptance Ratio is calculated which actually means that no. of times the system has reported true user as an imposter and no' of times the system has reported an imposter as true user respectively. These FAR and FRR should be as low as possible for a successful model and the project. There are ways that can help keep these parameters as low like by increasing the no' of feature extraction simply increases the classification model's dataset and hence, more ways can be figured out to keep out an imposter. We tested our system with various parameter values in the trust model.

The used trust model algorithm is as follows:

1. Collect user’s data or simulate the data keeping a perfect seeding bar.
2. Keep the typing pattern as unique and store it somewhere.
3. Now collect data for an imposter and keep it different from the true user.
4. Now train the model using both types of datasets.
5. Now try verifying the model using an example dataset or try inputting physically and let the classifier report you as an imposter and a true user.

In the classification one can use any classification algorithm we used z-score and also support vector machine classification algorithm for classifying the user as a true user or as an imposter. A global threshold was not able to produce and output good results but when using a local threshold, the results and classification was better. The main purpose was to not report a genuine user as an imposter and to not report an imposter as a genuine user i.e. the FAR and FRR are to be kept low as possible. This could be done using the shift key analogy and dividing the keyboard into two halves such that the features are extracted using the left hand for the left half keyboard and for the right hand the right half of the keyboard is used. The main purpose was to have a personal threshold where the genuine user was never locked out of the system.

3.2 Analysis method

The working of the system is divided into two main phases:

I. Training Phase: During the training phase the user are asked to input a particular username and password for a given number of times and these users are said to be true users. The imposter dataset is also made by inputting the data from a false user then the model is trained i.e. the classification is set ready.

II. Testing Phase: During the testing phase a random user tries to log in and the datapoints i.e. the typing pattern is taken from that random user and compared with the stored data of true users and imposters. Then that random user is classified as a true user or a false user or an imposter.

In the testing phase the performance of the system be measured in terms of Average Number of Genuine Actions, ANGA and Average Number of Impostor Actions, ANIA.

The genuine user plots almost lie on top of each other while that of an imposter differ from each other very much. This makes easy for the classification algorithm to note the difference in the patterns and classify it as an imposter or true user. In the end the False Rejection ratio and False acceptance Ratio is calculated which actually means that no. of times the system has reported true user as an imposter and no' of times the system has reported an imposter as true user respectively. These FAR and FRR should be as low as possible for a successful model and the project. There are ways that can help keep these parameters as low like by increasing the no. of feature extraction simply increases the classification model's dataset and hence, more ways can be figured out to keep out an imposter. This means that if the user's typing is in accordance with the classifier model, then the trust in the genuineness increases, otherwise it will decrease. The data samples from the genuine user will in most be a high probability from the classifier and sometimes a low probability. This means that most often the trust increases and sometimes it decreases. This then results in a trust value that remain at a high level. For impostor users this situation is the opposite. The simulation of the genuine user is not impossible and whenever it could be done the trust score will decrease. The general trend for the trust value will be downwards, and once the trust value reaches below threshold, then the user will be locked out. ANIA and ANGA are two parameters that are similar to FAR and FRR and they stand for Average number of imposter action and Average number of genuine actions. The Average number of genuine Actions should be as high as possible for a system to be successful and the Average number of Imposter actions must be as low as possible. The imposter is to be detected very quick so as to reduce the threat of losing any important files out in the open or public access.

3.4 Proposed System

The proposed prototype model has two different phases or stages namely the Enrolment stage or the training stage and the Authentication stage or the testing stage.

- Enrolment Stage: In this stage the users are asked to input their username and passwords and their typing patters and the features are recorded as a CSV file. So as to use that in the classification time. Then some imposter users typing pattern is also input which helps in the classification time.
- Authentication Stage: At this stage, During the testing phase a random user tries to log in and the datapoints i.e. the typing pattern is taken from that random user and compared with the stored data of true users and imposters. Then that random

user is classified as a true user or a false user or an imposter. At last, the process of verification yields two types of action: accepted or rejected user access. The characters are extracted from the CSV file and the classification is used for showing the user as a true or a fake user.

The proposed system is given as Figure 1. The feature for keystroke dynamics i.e. the Dwell time and Flight time is extracted from the CSV files that were from the calculated and recorded timestamps. The negative value happens when the user before releasing the current key, user presses the next key or multiple keys are pressed. The keystroke duration is just composed by positive whole values, the Keystroke Latency can contain positive values as well as negative values. This generally happens when a client has an act of composing. The classifier is answerable for choosing validation. After getting any info, the client is acknowledged or dismissed, in view of Criterion of Separation for example edge by the Classifier that predicts the similitude between the example to be checked and the layout of the model taken.

The whole model is divided into four distinct steps. These are listed as follows:

1. An individual register their roll No. and password with the database. Then the user has to type his text and train the machine.
2. Features are extracted when individuals press and release keys and uses mouse. More specifically the delay between the key-down and key-up time and the mouse speed and click times.
3. The algorithm is applied while the threshold is generated based on the variations that the user has done while typing the training set and using the mouse.
4. Calculating the SVM between training and test sample to get the users' score.

3.5 Threshold Determine:

The threshold determination is what makes the model adaptive and different than other existing models and algorithms. The window for error is the space in which user is permitted to cause any errors. This is decided by a method called Leave One-Out Method i.e. LOOM. This method is explained below in some detail in steps:

1. Out of the n samples, divide the training space of n samples to a single sample which is used as test sample, and n-1 samples used to create the training sample.
2. Apply a distance measure like Euclidean in our model to calculate the distance between the selected test sample and the mean vector of the n-1 training samples.
3. Perform the step 2 for n times to produce different thresholds for each feature vector.

4. The average of the n thresholds is calculated which produces the one single threshold that would represent the effective measure for all the thresholds in total.

5. These steps are repeated for calculating the individual thresholds for the distances measure.

The threshold calculation and also the execution of the algorithm can be clearly stated as a visual representation as below. Assume that the space is made of six data points each of them which stands for a set of attribute array in the database. Now pick any data point at random and calculate the distance of that particular point from each of the rest of data points.

That would give the threshold that must exist for this particular data point. Now choose another data point and repeat the process. At the end we would end up with some different difference vectors. Now take the average of these vectors and conclude to a single point in space. This point is cumulative distance which is equivalent of all the vectors combined.

Now let us imagine a sphere centered at this point. The threshold which is calculated would be the radius of this sphere. If any test data point arrives, we plot this point in space. Then we check if this point is inside the so formed sphere. If it does, then it is equivalent to a data set which is of a valid user and its delay array is within bounds of error. If it doesn't, then it would mean that the data set belongs to that of an imposter or a false/fake user and the delay discrepancy is beyond the allocated margin.

3.6 Design and technology

Using the design that is described in the above section the experiment software will be successfully built, tested and used to gather data. Rather than give an in-depth analysis of the code, we shall provide a more informative overview of the technologies we used, and describe the interactions with the software that volunteers will experience.

We built the experiment software as prototype web app using the following:

1. HTML5: For front-end and creation of forms to accept the template data from the users.

2. JavaScript: Perform the front-end validations and also to gather the different keystroke times from the users and the mouse usage, click time from the users.
3. CSV: After performing all operation we store data into csv file.
4. Google’s Colab: Plot the graph of keystroke patterns for dwell time, flight time, typing speed typing patterns.

The software is designed to be modular. This means that if similar experiments are required, the software to very easily be re-configure to with different groups, passphrases and schedules. The volunteers will be authenticated with the site using their username and a password.

Once authenticated they will be directed to a page containing a JavaScript client which allowed them to perform the experiment.

3.7 Decision making/Mathematical:

The proposed system is evaluated using two statistics. These metrics are as follows:

1. False Rejection Rate i.e. FRR, which is the refused fraction of Genuine users.
2. False Acceptance Rate i.e. FAR, which is the accepted fraction of impostor/false/fake individuals.

Equations shows FRR and FAR respectively.

$$FRR = \frac{Number\ of\ refused\ genuines}{Total\ number\ of\ genuines}$$

$$FAR = \frac{Number\ of\ accepted\ imposters}{Total\ number\ of\ imposters}$$

The biometric system performance could be measured using Equal Error Rate i.e. EER which refers to a point on the ROC i.e. Receiver Operating Characteristics curve, where the FAR and the FRR are equal.

Here we are going to discuss the result that we have achieved from this research and the discussion related to the result. Although we can use six ML to evaluate the system, we will report here only the results of the SVM classifier. As mentioned before that some impostors not detected for each of the other five classifiers, even when using a personal threshold. For the other 5 classifiers the probability that an impostor user was not detected when using a personal threshold ranged from 12 percent for KNN to 76 percent for Decision tree learning. Therefore, we will focus only on the results obtained with SVM. We created some SVM classifier models, i.e. one for each user. For the performance analysis we calculated the ANGA and ANIA values for fixed lock out threshold and for personal threshold

Support Vector Machine (SVM) Classification

The Machine Learning module is the part of a system that receives feature vectors of the users and performs classification process. Based on testing which is described forth, the Support Vector Machine i.e. SVM and Logistic Regression Algorithms i.e. LRA were chosen for the implementation. In order to implement these algorithms, we used the Sklearn library in python modules. It has been briefly introduced with the usage in the previous module. This module contains the Classification classes. Inside those classes, two methods are introduced, train and predict. As the names suggest, one method is for training the model with algorithm while the other one is used for prediction of classification decisions.

As illustrated, the algorithm receives the features and labels from the previous modules and fits a model using them. In order to perform classification, algorithm receive samples from previous modules and performs classification, as well as returns a probability for it. The implementations of the two algorithms are very similar. The main difference is the way the model is initialized for the processes.

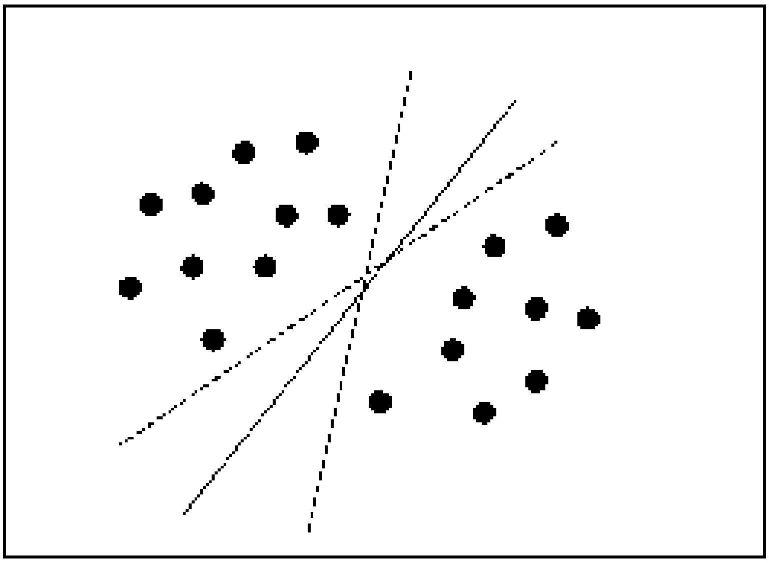


Figure 6: Support Vector Machines Classification

Supervised learning and unsupervised learning are used for operating in two types of environments for continuous authentication-based systems i.e. open-setting environment and the other is closed setting environment. Int the closed setting environment the user can be verified as an imposter or as a true user as the user's profile and imposters profile data are already known to the system. But this is not the case with an open-setting environment system, there the user's profile and the imposter's profile are not known in advance instead it works as the profiles of the user and the imposter builds and constantly keeps classifying the user as a true user or an imposter. Our framework will work in an open-setting condition situation. It will have applications in group based unlabeled information characterization frameworks. Irregularity Detection alludes to identification whether an example got affirms to the profile, is an inlier, or is an oddity to that profile's information, or is an anomaly. Recognizing clamor and abnormality is one of the issues in this procedure when we effectively attempt to perceive irregularities. It has an application in Intrusion and Fraud Detection Systems. Our Approach for the proposed framework is an open-setting condition, where with the utilization of unaided learning and abnormality recognition, we attempt to group approaching information is a client profile or a peculiarity, i.e. intruder/fake.

There are several steps/Approach/test plans to build system & the obtained results can be divided into the workflow phases as follows, shown in Figure:

- Log in process
- Data Collection
- Features Extraction

- Profile Creation
- Intruder Attack Simulation or random user entries.
- FAR & FRR Calculation

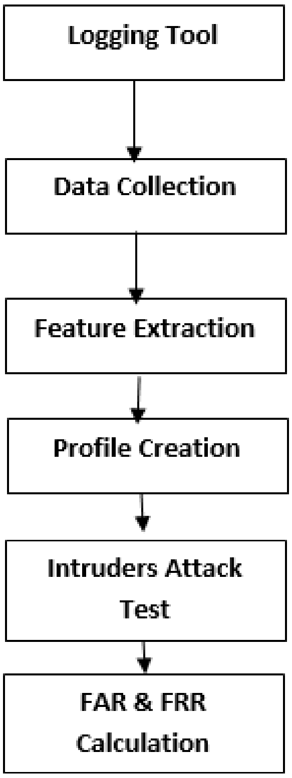


Figure 7: Prototype model overview

A. Logging process

First, we create a login form in html which asks for username and password and then it records the timestamps as key-press and key-release, also the usage of shift key is also recorded. Later this recorded data is used to calculate the dwell time, the flight time and typing speed of the users. The imposters are also allowed to login so as to record their typing patterns.

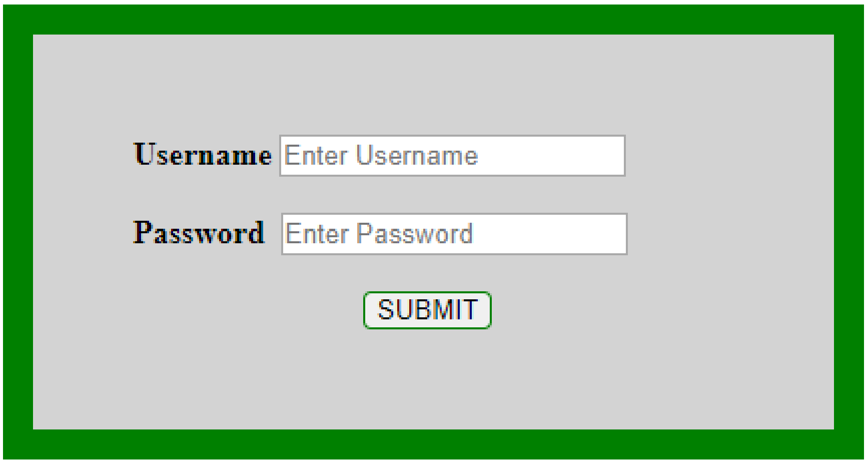


Figure 8: Login Window for the user's data intake for keystrokes.

B. Data Collection

The data can be collected by letting the users physically type the username and passwords to record the timestamps or one can simulate over for a dataset generation and to use it later. If simulation is used then proper seeding has to be used because the genuine users typing patterns have to be matching and not to simply be random.

C. Feature Extraction

After the data of the timestamps gets collected and the CSV file gets downloaded or updated. The different function can calculate the features of the Keystroke Dynamics that are Dwell time, Flight Time and typing speed. One may also choose these features to be extracted right from the beginning like from the dataset generation, but it will be time taking rather it should be preferred to be done later.

D. Profile Creation:

The calculation gets the highlights and names from the past modules and fits a model utilizing them. So as to perform grouping, calculation get tests from past modules and performs arrangement, just as returns a likelihood for it. The executions of the two calculations are fundamentally the same as. The fundamental distinction is the manner in which the model is introduced for the procedures. The SVM calculation maps the focuses in space with the goal that the instances of the different classifications are partitioned by a reasonable hole that is as wide as could reasonably be expected. New models are then anticipated to have a place with some class, in light of which side of the hole they fall on. Our Approach for the proposed framework is an open-setting condition, where with the utilization of solo learning and peculiarity location, we attempt to group approaching information is a client profile or an irregularity, for example true/fake. But, in one-class SVM, the support vector model is prepared on information that has just one class, which is the "ordinary" class. It separates the properties of "typical" class and from these properties can figure which models resemble the ordinary class or unique in relation to it. This is useful for peculiarity location because of the scarceness of preparing models is the thing that characterizes oddities. Methodology for Profile Creation: First, we imported the .csv group into a Data Frame. At that point we standardized the information, so all qualities lie in the scope of - 1 to +1. This is done to improve the presentation of SVM. Then using the Sci-kit-learn library's One-Class SVM, we created the profile using RBF Kernel with nu=0.5 & gamma=0.00005.

E. Intruder Attack Simulation

One can utilize a reenactment for a gatecrasher's assault however I did the interloper's assault with a portion of different companions. We had chosen a few members whose gathered information were of size enormous enough to work with. We made the profile for these 4 Users with the assistance of One-Class SVM. At that point for every client, we utilized the 3 outstanding member's key information, and 1 other member key information as gatecrasher over their profile.

CHAPTER 4

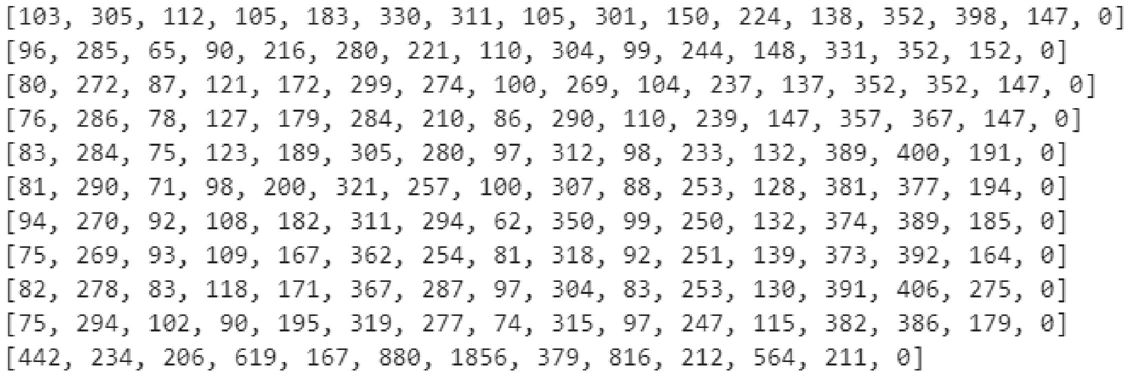
Performance Analysis

Feature Extraction

After collection of data from the participants, we extracted Dwell Time & Flight time from the raw files in .csv format.

Dataset

Example Format 1: Here the dataset length in every line i.e. every attempt is same except the last one which is an input from the imposter.



```
[103, 305, 112, 105, 183, 330, 311, 105, 301, 150, 224, 138, 352, 398, 147, 0]
[96, 285, 65, 90, 216, 280, 221, 110, 304, 99, 244, 148, 331, 352, 152, 0]
[80, 272, 87, 121, 172, 299, 274, 100, 269, 104, 237, 137, 352, 352, 147, 0]
[76, 286, 78, 127, 179, 284, 210, 86, 290, 110, 239, 147, 357, 367, 147, 0]
[83, 284, 75, 123, 189, 305, 280, 97, 312, 98, 233, 132, 389, 400, 191, 0]
[81, 290, 71, 98, 200, 321, 257, 100, 307, 88, 253, 128, 381, 377, 194, 0]
[94, 270, 92, 108, 182, 311, 294, 62, 350, 99, 250, 132, 374, 389, 185, 0]
[75, 269, 93, 109, 167, 362, 254, 81, 318, 92, 251, 139, 373, 392, 164, 0]
[82, 278, 83, 118, 171, 367, 287, 97, 304, 83, 253, 130, 391, 406, 275, 0]
[75, 294, 102, 90, 195, 319, 277, 74, 315, 97, 247, 115, 382, 386, 179, 0]
[442, 234, 206, 619, 167, 880, 1856, 379, 816, 212, 564, 211, 0]
```

Figure 9: The example of dataset of ten samples for one user

In the Extraction of Dwell Time & Flight Time from the raw log files, we needed to take care of noise values

As shown in figure below the red lines show the pattern of a true user logins which matches with all the other login patterns or maybe we can say all the patterns almost are identical hence we infer that the user is not an imposter. But the black line pattern is very different from the one of the red lines hence we can say that pattern is of an imposter. It is worth noticing that the black line is shorter than that of red lines hence we can infer that the imposter either didn't used the shift keys and went for caps-lock for capital letters or he used tabs key instead of clicking on the input field.

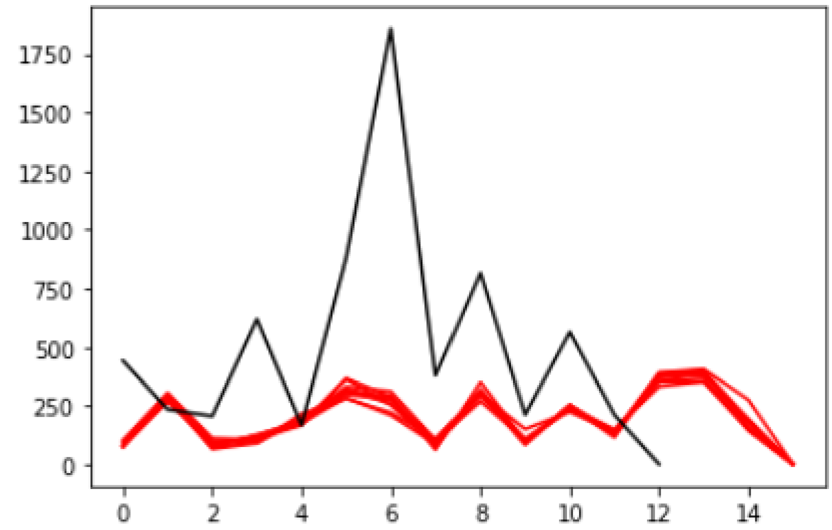
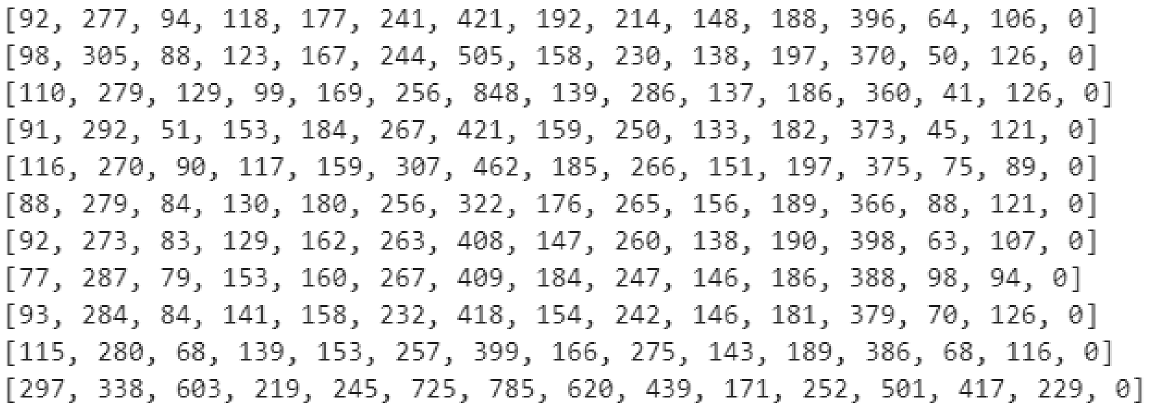


Figure 10: The plot of the typing pattern of a user vs an imposter.

Example Format 2: Here the dataset length in every line i.e. every attempt is same hence no one used any extra tabs or anything different.



```
[92, 277, 94, 118, 177, 241, 421, 192, 214, 148, 188, 396, 64, 106, 0]
[98, 305, 88, 123, 167, 244, 505, 158, 230, 138, 197, 370, 50, 126, 0]
[110, 279, 129, 99, 169, 256, 848, 139, 286, 137, 186, 360, 41, 126, 0]
[91, 292, 51, 153, 184, 267, 421, 159, 250, 133, 182, 373, 45, 121, 0]
[116, 270, 90, 117, 159, 307, 462, 185, 266, 151, 197, 375, 75, 89, 0]
[88, 279, 84, 130, 180, 256, 322, 176, 265, 156, 189, 366, 88, 121, 0]
[92, 273, 83, 129, 162, 263, 408, 147, 260, 138, 190, 398, 63, 107, 0]
[77, 287, 79, 153, 160, 267, 409, 184, 247, 146, 186, 388, 98, 94, 0]
[93, 284, 84, 141, 158, 232, 418, 154, 242, 146, 181, 379, 70, 126, 0]
[115, 280, 68, 139, 153, 257, 399, 166, 275, 143, 189, 386, 68, 116, 0]
[297, 338, 603, 219, 245, 725, 785, 620, 439, 171, 252, 501, 417, 229, 0]
```

Figure 11: Example dataset of a user

The pattern looks like this, for comparing out of all the input attempts.

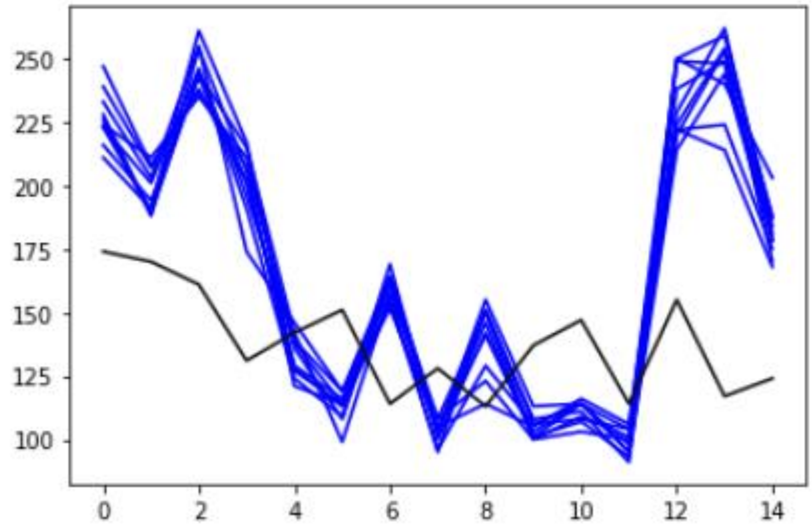


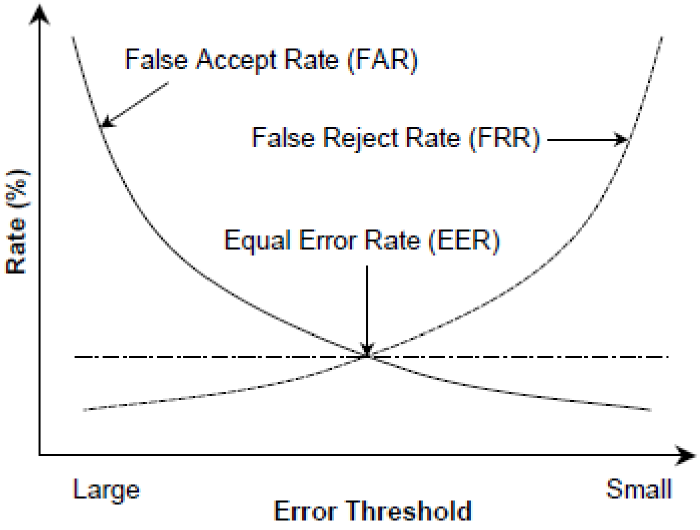
Figure 12: The typing pattern of another user vs an imposter.

MATCHING PROCESS

We compute the distinction and characterize the dataset as a genuine client and a bogus client. At that point we compute the False acknowledgment proportion and the False dismissal proportion and these ought to be as low as conceivable which makes the framework increasingly fruitful. We contrast this incentive with the present time of login client if the worth will be coordinate as per the edge esteem the individual acknowledged or called confirmed client. This worth will be change as indicated by examination. Using these outputs we can calculate FAR (False accept ratio) and FRR (false Reject Ratio) values. These thresh hold values increases the efficiency of result.

F. FAR & FRR Calculation

We take the user data multiple time for a limited range and after the data input we



FAR Results

We accept the limit an incentive to contrast and time. These limit esteems are expanded the proficiency of result. We contrast this incentive with the present time of login client if the worth will be coordinate as indicated by the limit esteem the individual acknowledged or called verified client. This worth will be change as indicated by investigation. Utilizing this we will compute FAR (False acceptance Ratio) and FRR (False Reject Ratio) values. Utilizing reproduction or genuine client's information will give a FAR, so at the testing/validating stage the FAR ought to be as low as conceivable this shows the accomplishment of the framework. These reenacted information sections of the client's information that were acknowledged even as bogus will fall in this FAR

FRR Results

Utilizing reenactment or genuine client's information will give a FRR, so at the testing/confirming stage the FRR ought to be as high as conceivable this shows the accomplishment of the framework. These reproduced information sections of the client's information that were dismissed as bogus will fall in this FRR.

Average FAR & FRR of System

We accept the limit an incentive to contrast and time. These limit esteems are expanded the proficiency of result. We contrast this incentive with the present time of login client if the worth will be coordinate as indicated by the limit esteem the individual acknowledged or called verified client. This worth will be change as indicated by investigation. Utilizing this yield we will compute FAR (False acknowledge proportion) and FRR (bogus Reject Ratio) values.

CHAPTER 5

CONCLUSION

In this project I have used the user’s typing pattern for authenticating the user as true user or not a true user. The typing patterns of the true user will differ from that of any imposter who comes to know user’s password as well as the username. The true user will remember his/her username as well as the password as well as the typing pattern. It increases the security of the system as no. of layers have increased.

The FAR can be decreased if we increase the no. of extracting features. Like introducing “Shift key Analogy”, in this one can also take into account which shift key user is using like if the system keyboard has two shift keys. Also, if one will filter out the keyboard into two halves and work with the analogy that “nearer keys will have lesser flight time”.

There are advantages of using CABB. The data collected is almost impossible to auto-generate. There is not requirement of any extra hardware. The collected data is stored in the user’s device so, there is no breach of privacy.

Regarding the areas of applications for this project are a place where attendance is required and one cannot use a camera then if a user tells a friend their password then he/she would not be able to mark the attendance as the typing patterns will differ significantly. Other than this anywhere when we cannot use a camera or any other security layer more advanced than we can go with CABB.

CHAPTER 7

REFERENCES

- S. Abe, Support Vector Machines for Pattern Classification. New York: Springer, 2005.

- A. A. E. Ahmed and I. Traore, “A new biometric technology based on mouse dynamics,” IEEE Trans. Depend. Secure Comput, vol. 4, no. 3, pp. 165–179, Jul./Sep. 2007.

- Chao Shen , Zhongmin Cai, Xiaohong Guan, Youtian Du, State Key Laboratory for Manufacturing Systems, Xi'an Jiaotong University, China Roy A. Maxion, Dependable Systems Laboratory, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA

  IEEE Transactions on Information Forensics and Security

  Volume: 8 , Issue: 1 , Jan. 2013

- Mahnoush Babaeizadeh, Majid Bakhtiari, and Mohd Aizaini Maarof, “Authentication Method through Keystrokes Measurement of Mobile users in Cloud Environment”, Int. J. Advance Soft Compu. Appl, Vol. 6, No.3,November 2014 ISSN 2074-8523

- Y. Zhong and Y. Deng, Recent Advances in User Authentication Using Keystroke Dynamics Biometrics DOI: 10.15579/gcsr.vol2.ch3, GCSR Vol. 2, pp. 41-58, 2015

- D. C. D’Souza. “Typing Dynamics Biometric Authentication.” Oct. 2002. http://innovexpo.itee.uq.edu.au/ 2002/projects/s373901/thesis.PDF